Andres Concecion
Summer 2009 DREU

The summer DREU project at Brooklyn College consisted of two main parts. The first part was working on a soccer field integration into the Player/Stage robotic simulation and the second part was the Surveyor Robot 3D to 2D mapping project.

The Soccer field project used the Player/Stage program to attempt to create a working soccer field in which the simulations can run in and play a game of soccer. Player/Stage is a program that allows a user to simulate a robot in a 2 dimensional map that the robot can interact inside. This allows a user to simulate what a robot will do instead of having to run that robot in the real world. Player/Stage is a client-server model that generates a player in the stage environment and allows the user to send command to that player via a terminal window where the user can input commands.

The project I was working on was trying to implement a system where multiple robots could play a real game of soccer. The first step I took was an attempt to figure out how to get a real soccer field on the stage environment that the players could recognize and interact with, at least with the goals and edges. Before any of that could be accomplished I needed to familiarize myself with player and stage, finding out how the code works. Stage allows a user to take a .png file and bitmap that image file into the stage environment. I managed to isolate the colors that stage picked up as obstacles (anything in the blue/green spectrum). This lead me to create a test image, basically color bars, that would let me see what each color was made into in the stage environment. I found that creating a soccer field would be nearly impossible since the stage environment only considers objects, like walls and rooms, and anything not an object is left blank and considered part of the floor. I couldn't get a working soccer field with all the markers but I was able to get a room with two goals on either end that a robot can navigate through.

The soccer ball implementation was not very successful, I was able to attain a paper on how someone had implemented a ball in the stage environment but I was not able to find any source code online or via contacting the person who wrote the paper. I attempted to write the code myself for the ball but wasn't able to put together the class with all of the physics equations that were required to make a real ball.

Player/Stage is built so that a user can have multiple robots, or players, on the same stage working together. In order to create a cohesive robot team, I needed to be able to send one command automatically to all the players on a team at the same time. Right now player/stage allows you to send a command via a terminal for each individual player. However, a user can specify which port he would like to send the command to, so I thought that there was a way to be able to send one command from one terminal and have every robot get it at the same time. First I had to get the player/stage code in a working user directory in my lab machine so I can make changes and see the changes when I run the program and also have a global version as a backup should I change something really drastic in the code. My professors where able to help me get a local version of the code working on my lab machine and I spend quite a while reading code and documentation to figure out how to make my simultaneous commands work.

I was able to determine that what I planned was theoretically feasible, I just had to change the entire structure of how commands where sent in player/stage. The original way that commands were sent was by a typical class creation from the information that the user provided. In order to make my method work, I needed to create multiple structures for each robot and send a command out through each structure. Each structure was going to be allocated in memory first and then instantiated in a loop. Then came several huge issues with the classes that I tried to change, having to create new empty and copy constructors and new methods to set information and get information. The implementation then was unable to compile due an issue with the newer kernel running on the machine and the methods used by stage in some of its structs. I contacted the author of the code about my issue and his suggestion to revert to an older kernel did not work on my machine.

After the trip to Austria for the RoboCup competition we were given a new task and a new toy to play with, the surveyor robot. My task was to ultimately have the robot run through a course and be able to change the 3D image it was seeing into a 2D, top down view map of the course. This is known and simultaneous localization and mapping (SLAM). The surveyor robot is a small, two thread wheel robot equipped with a single web-cam style camera on top and an infrared sensor on each side. After trying get workable readings from the infrared sensors, I contacted one of the other professors that use to work with the surveyor. He told me that they IR sensors were too imprecise for any workable applications. Without any distance measurements, the SLAM task with single camera became much harder.

Luckily, after much research there were a few papers that seemed to show some promise in SLAM with a single camera. MonoSLAM is program written by Andrew Davison that utilizes a single camera to obtain information about the environment to create a 2D map. His paper "SLAM with a single camera" described that by using several 2D images one can use vector mathematics to partition the images to create a 3D model of a particular object and insert it into the map. The source code for the MonoSLAM program had issues compiling on the lab machine. After changing the makefile in order to use the older kernel that was on the lab machine the code still would not compile. I did spend some time looking over the code and copying several sections to parse some images but could not get the code to compile without using all of the dependencies that come in the program.

All the work that I did on the MonoSLAM and player/stage is saved on the lab machine with specific instructions on what changes I made. I hope that I can work on this project further, as there are many things that I left unfinished. The player/stage modifications create a good base for soccer simulations and multi-bot rescue missions. The surveyor project is something that really sparked my interest in robotics after witnessing some really interesting robots at the RoboCup 2009. I hope to continue my education in robotics with Prof. Sklar at Brooklyn College, it was one of the most rewarding educational experiences of my life.

Bibliography:

-″Real-Time Simultaneous Localisation and Mapping with a Single Camera "
Andrew J. Davison
Robotics Research Group, Dept. of Engineering Science, University of Oxford, OX1 3PJ, UK
ajd@robots.ox.ac.uk
http://www.robots.ox.ac.uk/ ãjd/
-"The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems".
Brian Gerkey, Richard T. Vaughan and Andrew Howard
In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003)*, pages 317-323
Coimbra, Portugal, June 2003.
-"Player 2.0: Toward a Practical Robot Programming Framework".
Toby H.J. Collett, Bruce A. MacDonald, and Brian P. Gerkey
In *Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005)*
Sydney, Australia, December 2005.
-Simultaneous Localization and Mapping: Parts I and II

by Hugh Durrant-Whyte and Tim Bailey